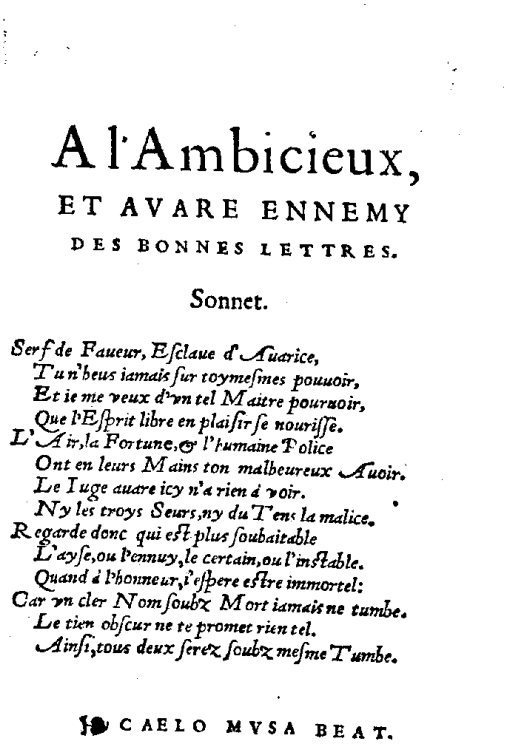
Un peu de XPath, un peu de XSLT

Lou Burnard and Alexei Lavrentev *2022-06*

1 Exercice XPath et XSLT



A l'Ambicieux,
ET AVARE ENNEMY
DES BONNES LETTRES.

Sonnet.

Serf de Faueur, Eſclaue d'Auarice,
Tu n'beus iamais ſur toymeſmes pouuoir,
Et ie me veux d'vn tel Maitre pouruoir,
Que l'Eſprit libre en plaiſir ſe nouriſſe.
L'Air, la Fortune, & l'humaine Police
Ont en leurs Mains ton malheureux Auoir.
Le Iuge auare icy n'a rien à voir.
Ny les troys Seurs, ny du Tens la malice.
Regarde donc qui eſt plus ſoubaitable
L'ayſe, ou l'ennuy, le certain, ou l'inſtable.
Quand à l'honneur, i'eſpere eſtre immortel:
Car vn cler Nom ſoubz Mort iamais ne tumbe.
Le tien obſcur ne te promet rien tel.
Ainſi, tous deux ſerez ſoubz meſme Tumbe.

CAELO MVSA BEAT.

Dans ce petit exercice et le suivant, on prend comme exemple une version numérisée d'un célèbre sonnet de DuBellay. Vous trouverez une version de ce sonnet déjà balisée dans le fichier duBellay.xml — ouvrez-le d'abord avec Oxygen.

2 XPath

2.1 Navigation avec XPath

oXygen est très pratique pour voir un peu ce que l'utilisation de XPath seul peut apporter à un processus d'exploration, d'appropriation ou de vérification de document TEI. La ligne de saisie "XPath" que l'on voit a gauche en haut de la fenêtre, au-dessous des barres d'outils oXygen, permet de saisir des expressions XPath. Après avoir tapé sur la touche Entrée au clavier, oXygen évalue l'expression et retourne, dans la fenêtre qui fait toute la largeur de l'écran en bas, une liste de(s) résultat(s) – un ou plusieurs ensembles de nœuds, en donnant pour chaque nœud résultat l'expression XPath qui permet de l'atteindre. Si on clique avec la souris sur une des lignes de résultat, oXygen montre le nœud résultat sélectionné dans son contexte TEI au sein de la fenêtre de l'éditeur.

Voici une petite liste de requêtes que vous devriez essayer d'exprimer au moyen du langage XPath :

1. quel est l'élément racine du document XML ?
2. quel est le titre donné à ce document ?
3. qui est le responsable de la création du fichier ?

4. quelle est l'origine de la version encodée ?
5. quelle est la date de la révision la plus récente du document ?
6. combien y a-t-il de strophes (`<lg>`) dans le sonnet ?
7. combien y a-t-il de segments de texte dont la graphie est considérée incorrecte (`<orig>`) aujourd'hui ?
8. combien y a-t-il de vers (`<l>`) qui contiennent au moins un tel segment ?
9. quel est le contenu textuel du premier vers de chaque strophe ? et du dernier?
10. Quels nœuds texte contiennent la chaîne de caractères "jamais"

On va faire ces exercices ensemble. Des réponses sont proposées dans la section qui suit : ne les regardez pas si vous préférez tester vos compétences !

2.2 Exercice XPath corrigé

Quel est l'élément racine du document XML ? Expression XPath à écrire :

`/*` [le nœud document est désigné par le slash ; la 2^{e} étape désigne les enfants de type élément de ce nœud]

Réponse : TEI

Quel est le titre donné au document ? Expression XPath à écrire :

`/TEI/teiHeader/fileDesc/titleStmt/title` ou encore (entre autres possibilités) : `//titleStmt/title`

Réponse : On voit les trois nœuds `<title>` qui répondent à la question. Si on voulait obtenir le titre proprement dit (sans le complément de titre), il faudrait connaître précisément le modèle auquel obéit ce document, et demander le nœud `<title>` en première position (//titleStmt/title[1]), ou de choisir le nœud `<title>` dont l'attribut *@type* porte une valeur appropriée (`//titleStmt/title[@type='main')]`).

Qui est responsable de la création et de l'édition du fichier ? Expression XPath à écrire :

`//titleStmt/respStmt/name`

Réponse : 2 nœuds ici.

Quelle est l'origine de la version encodée ? Expression XPath à écrire :

`//sourceDesc/*` ou, plus finement, `/TEI/fileDesc/sourceDesc/*`. Sans le * vous ne recevrez que l'élément de groupement `<sourceDesc>`, sans ses composants.

Quelle est la date de la révision la plus récente du document ? Expression XPath à écrire :

`revisionDesc/change[1]/@when`. Par défaut, le premier `<change>` devrait être le plus récent, et son attribut *@when* donnera sa date dans un format normalisé.

Réponse : 2017-10-18

Combien y a-t-il de strophes (`<lg>`) dans le sonnet ? Expression XPath à écrire :

`count(//lg)`. On se sert de la fonction `count()`.

Réponse : 4

Combien y a-t-il de `<orig>`, i.e. des segments de texte dont la graphie est considéré incorrecte
Expression XPath à écrire :

`count(/TEI/text//orig)`. Enlevez le `count()` si vous préfériez les voir…

Réponse : 18

Combien y a-t-il de vers (`<l>`) qui contient au moins un tel segment? Expression XPath à écrire :

```
count(//l[descendant::orig])
```

Réponse : 7

Et combien y a-t-il de `<orig>` qui sont contenu directement par un `<l>` ?
Expression XPath à écrire :

```
count(//l/orig)
```

Réponse : 3

Quel est le contenu textuel du premier vers de chaque strophe ? et du dernier ?
Expression XPath à écrire :

```
//lg/l[1]//text()
```

Pour le dernier, c'est un peu moins évident… `//lg/l[position()=last()]//text()`

Quels nœuds texte contiennent la chaîne de caractères "jamais" ? et quels vers?
Expression XPath à écrire :

```
//text()[contains(., 'avare')]
```

```
//l[contains(., 'avare')]
```

Réponse : la première liste de réponses contient 2 nœuds; la deuxième qu'un.

3 XSLT

Dans ce deuxième petit exercice, nous allons faire des expériences avec XSLT sous contrôle d'oXygen. On va travailler avec cette version du même sonnet de DuBellay qui a été (partiellement) enrichie avec des balises `<choice>`.

3.1 Transformation en HTML

L'application classique de XSLT est de transformer un document TEI XML en HTML pour le visualiser dans un navigateur web. On commence donc par là.

Pour faire cela, il faut mettre en place une liaison entre le fichier XML d'entrée, le fichier HTML de sortie, et le fichier XSLT qui va contrôler la transformation. Il y a une méthode plus élaborée pour créer ce qu'on appelle un *scénario de transformation*, mais pour l'instant nous allons effectuer cette liaison de la manière la plus simple.

1. Ouvrez Oxygen. Sélectionnez Ouvrir sur le menu Fichier et ouvrez le fichier duBellay.xml.

2. Sur le menu Document sélectionnez Document XML, et ensuite Associer une feuille de style XSLT/CSS ou cliquez le bouton avec un épingle bleu sur la barre d'outils

3. Dans le dialogue qui s'affiche, assurez vous d'ouvrir l'onglet XSLT. Puis tapez `duBellay.xsl` dans le champ URL.

4. Votre document duBellay.xml est maintenant doté d'une *instruction de traitement* `<?xml-stylesheet type="text/xsl" href="duBellay.xsl"?`

5. Sur le menu Document sélectionnez Transformation, et ensuite Appliquer le/les scénario(s) de transformation ou tapez `ctrl-maj-T`, ou bien cliquez sur le bouton avec un grand triangle rouge sur la barre d'outils.

6. En bas de l'écran, le message `Transformation réussie` apparaît…

7. .. et après un bref délai, votre navigateur devrait s'ouvrir pour afficher le fichier duBellay.html que vous venez de créer.

8. C'est joli, hein ? Peut-être y a-t-il encore un peu de travail à faire…

3.2 Une feuille de style XSLT

Une feuille de style XSLT est un document XML comme les autres. Il peut donc être édité avec Oxygen.

- Allez de nouveau sur Fichier - Ouvrir. Ouvrez le fichier duBellay.xsl dans votre dossier de travail.

- Ce fichier XSLT contient un seul template, qui correspond à un élément nommé `<TEI>`, l'élément racine de notre document. Son effet sera de produire un nouveau document XML, dont l'élément racine sera nommé `<html>`, et qui contiendra deux autres éléments: un `<head>` et un `<body>`.

- *Attention aux espaces de noms!* Nous n'avons pas explicité qu'il s'agit d'un 'TEI' de l'espace de noms de la TEI – mais nous avons précisé que ce dernier sera l'espace de noms par défaut, donc cela ne pose aucune problème.

- Dans le `<head>`, on met des métadonnées concernant le document HTML (cela correspond en quelque sorte au `<teiHeader>`); en l'occurrence nous fournissons par défaut un élément `<meta>` pour assurer l'affichage correct des caractères Unicode. Dans le `<body>` on va mettre le contenu de notre document, en appliquant la commande `<xsl:apply-templates>`

- Ce dernier va essayer d'appliquer tous les templates disponibles… il n'y en a pas, donc il ne va produire que le texte du document.

- … et en effet, si vous revenez sur votre navigateur et regardez la source HTML du fichier, c'est ce qu'il a fait.

Au travail ! D'abord, nous allons extraire de l'en-tête un titre pour le document HTML, et supprimer le reste du TEI Header.

- Ajoutez un élément `<title>` à l'intérieur de l'élément `<head>`, juste après la fin du `<meta>`. Pour trouver le titre du document il faut naviguer de la racine du document source (l'élément TEI) jusqu'à l'élément `<title>`, qui se trouve dans le `<titleStmt>`, dans le `<fileDesc>`, dans le `<teiHeader>`. Cela s'effectue avec un *XPath*, bien-sur !

- Tapez `<xsl:value-of select="teiHeader/fileDesc/titleStmt/title"/>`.

- Modifiez la balise `<xsl:apply-templates>`, en ajoutant `select="text"`

- Cliquez sur le bouton Indentation. Votre feuille de style maintenant devrait ressembler à ceci :

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xpath-default-namespace="http://www.tei-c.org/ns/1.0"
  version="2.0">

<xsl:template match="TEI">
  <html>
    <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"></meta>
      <title><xsl:value-of select="teiHeader/fileStmt/titleStmt/title"/></title>
    </head>
    <body>
      <xsl:apply-templates select="text"/>
    </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

- Cliquez sur l'icône de la disquette (ou tapez ctrl-s) pour enregistrer les modifications que vous venez de faire.

- Cliquez sur l'onglet duBellay.xml pour revenir sur votre document XML.

- Cliquez sur l'icône triangle-rouge pour relancer la transformation (ou tapez CTRL-maj-T)

- Voyons ce que cela donne : cette fois, on ne voit que le `<text>` de notre document. On fait des progrès !

Maintenant, on va ajouter des templates.

- Revenez sur la feuille de style.

- Ajoutez

  - un template pour `<l>`, qui va ajouter une balise `<br/>` après chaque vers
  - un template pour `<lb>`, qui va ajouter une balise `<br/>` à chaque saut de ligne
  - un template pour `<head>`, qui va l'entourer de `<h2>`
  - un template pour `<lg>`, qui va l'entourer de `<p>`, et en plus le préfixer du numéro de chaque strophe.

- Voici les templates requis… essayez de comprendre par vous-même le fonctionnement de chacun d'entre eux.

```
17  <xsl:template match="l">
18    <xsl:apply-templates/><br/>
19  </xsl:template>
20
21  <xsl:template match="head">
22    <h2><xsl:apply-templates/></h2>
23  </xsl:template>
24
25  <xsl:template match="lg">
26    <xsl:number/>
27    <p><xsl:apply-templates/></p>
28  </xsl:template>
29
```

3.3 Traitement des <choice>

Est-ce que le XSLT peut nous aider à mieux traiter ces éléments `<choice>`? Bien sur, oui. D'abord, on pourrait simplement ajouter un template qui va supprimer les `<orig>` (ou les `<reg>` si vous le préfériez).

Tout ce qu'il faut, c'est ajouter un template comme ceci : `<xsl:template match="orig"/>`. Essayez-le. Est-ce que vous comprenez comment cela fonctionne ? Si oui, vous avez bien compris les principes de XSLT !

3.4 Pour aller plus loin…

- Créez un paramètre modernisation avec `<xsl:param>`, avec comme valeur "oui" ou "non", selon votre préférence. Ceci va contrôler la suppression ou bien des `<reg>`, ou bien des `<orig>` selon la valeur qu'on lui donne. Par exemple, s'il aura la valeur `oui` on supprimera les `<orig>`.
- Ajoutez un template pour l'élément `<choice>`, qui contiendra un `<xsl:choose>` pour tester la valeur de ce paramètre
- Ensuite modifiez votre feuille de style pour entourer en guillemets la partie du `<choice>` affichée, pour signaler qu'il y a eu une modification de la source: Les experts en HTML peuvent proposer d'autre modes d'affichage éventuels pour les `<choice>`, sous contrôle d'autres valeurs du paramètre…

Afficher l'image du fac-similé (`pb/@facs`) de l'original avant la transcription

Qu'est-ce qu'on ferait pour n'afficher qu'une partie du texte? Modifiez votre feuille de style pour n'afficher que le vers initial de chaque strophe.

Et pour les plus ambitieux/ses, comment utiliser XPath pour créer une list des mots utilises dans le texte, chacun avec sa fréquence?

3.5 Pour tester votre compréhension

Le projet ELTeC met à votre disposition un ensemble de romans européens du 19ème siècle, tous balises dans un format TEI-XML minimal. Allez sur le site http://distantreading.github.io/ELTeC/fra/index.html pour voir la liste des titres français.

Choisissez un titre qui vous plaît, et cliquez sur l'icône TEI pour voir le texte en XML-TEI. Tapez `ctrl-a` pour sélectionner tout, et ensuite faire un copier-coller pour copier le document dans oXygen.

Notez plusieurs aspects du balisage: il y a un titre et un auteur; il y a une mention de la taille (en mots) du texte. Et si vous descendez un peu dans le fichier vous verrez que chaque chapitre est encodé avec un `<div type='chapter'>`, et commence avec un ou plusieurs `<head>` suivi des paragraphes encodés avec un `<p>`. Quelques uns des textes (mais pas tous) contiennent des vers, encodés avec un `<l>`; quelques un (mais pas tous) signalent un début de page dans la source avec une balise `<pb/>`. Avec vos connaissances de XPath et XSLT, il devrait être facile pour vous de créer un petit rapport indiquant:

- le titre et l'auteur
- le nombre des chapitres, des paragraphes, et des mots
- une liste des titres des chapitres
- une indication de la location du début du chapitre (ou bien comme numéro de page, ou bien comme numéro de paragraphe)
- une indication du nombre des vers, s'il y en a

Vous pouvez générez ce rapport en format texte brut, ou en HTML, comme vous voulez. J'ai fait le mien en texte brut, parce que je suis paresseux, et à partir du texte FRA00302, parce que j'aime les romans du wild west. Voici le début:

```
  Rapport sur "Belle rivière"  de Aimard, Gustave (1818-1883)

Ce roman contient 38 chapitres, 139314 mots, et 6284 paragraphes, soit en moyen
22.17 mots par paragraphe.

Voici une liste des chapitres, avec la location de son début, et le contenu de son
dernier paragraphe:
I -- Le comte de Jumonville... au paragraphe 1
Para final: Ils ne tardèrent pas à disparaître dans les fourrés et les taillis
épais qui servaient de remparts au camp des Français.
II -- Le major Washington... au paragraphe 264
Para final: Son cigare entièrement fumé, l'enseigne Ward s'enveloppa à son tour
dans son manteau et s'endormit du sommeil de Ponce-Pilate.
III -- L'assassinat... au paragraphe 437
Para final: Un silence funèbre plana alors sur cette place redevenue solitaire et
désormais maudite.
```

A vous de jouer!

3.6 L'anti-sèche

Vous trouverez dans le dossier Corrections nos propositions de réponse, dans les fichiers duBellayDisplay.xsl, duBellayLexique.xsl, et eltecRapport.xsl. Ne les regardez pas tout de suite !